

Projet en informatique musicale — MUS3325X

Musique Numérique

Rodolphe Busuttil-Décousus

- Plan de travail :

Début du projet : Janvier 2017

Fin du projet : septembre 2018

- Mise à jour prévue : chaque semaine

Rapport des mises a jours :

- 1ère entrée : 23 janvier 2018 (Mise en place de la mise en page du document, finalisation du Setting de Github)
- 2eme entrée : 24 janvier 2018 (Mise en place de l'énoncé et de la description.)
- 3eme entrée : 2 février 2018 (Découverte de **Automatonism 2.0** de **Johan Eriksson**, qui me permet de construire un synthétiseur modulaire.)
- 4eme entrée : 9 février 2018 (Création d'une ébauche de la patch du synthétiseur modulaire)
- 5eme entrée : 23 février 2018 (Installation de la librairie **GEM**, et début de l'exploration pour concevoir la patch visuel)
- 6eme entrée : 3 mars 2018 (Création d'une patch visuel **GEM**, mais remarque des problèmes d'optimisation, et de latence, sur la patch visuel autant que sur la patch audio.)
- 7eme entrée : 10 mars 2018 (Recherche d'une librairie supplémentaire associé à **GEM** qui me permettra de gagner du temps sur l'intégration de video, grâce à **Pixel TANGO**, librairie de la **SAT**(société des arts technologique de Montreal)

- 8eme entrée : 23 mars 2018 (Abandon de la librairie **pixel TANGO** et de **GEM**, a cause des problème d'optimisation et de latence, ainsi que la non mise a jour récente de ces dites librairies)
- 9eme entrée : 1 avril 2018 (Recherche d'une nouvel methode pour crée une patch visuel audio réactive, qui se soumet au moins de latence possible)
- 10eme entrée : 9 avril 2018 (Mise en place de la librairie **Ofelia**, qui effectue son rendu graphique à travers **OpenFramwork**)
- 11eme entrée : 15 avril 2018 (Refonte complète de la patch audio, pour en augmenté la facilité d'utilisation et le paramétrage de l'audioréactif)
- 12eme entrée : 20 avril 2018 (Création de la patch visuel et de son optimisation, pour fonctionné en harmonie avec la patch audio. Utilisation et modification des patches d'exemple et d'apprentissage d'**Ofelia** pour en faire une création original et facile de compréhension pour les Users)
- 13eme entrée : 25 avril 2018 (Finalisation de l'audioréactif et de ses paramètres à travers les deux patches, audio et visuel)

• Énoncé :

Outil modulaire de composition et de performance vidéo musicale.

Le titre de ce projet est en cours de précision, mais je trouve qu'il reflète l'idée d'ensemble de la réalisation.

• Description :

Le but de ce projet est de créer un outil qui permettrait de réunir en une seule plateforme (interface numérique) un instrument modulaire, qui comprendrait de la synthèse ainsi que des séquenceurs et des effets audio numériques.

Il sera également capable de générer des effets sur un generateur vidéo qui affichera de manière synchronisée en tout temps de la vidéo mise en relation avec le son.

Cette fenêtre vidéo fait partie intégrante de cet instrument, je veux ici concevoir un outil qui repose fondamentalement sur les deux aspects de traitement, audio et vidéo, une sorte de synthétiseur interactif (à la vidéo).

Cet outil comportera donc un synthétiseur modulaire, des traitements en temps réel pour l'audio et la vidéo.

Je voudrais contrôler l'ensemble des paramètres de l'interface numérique grâce à un contrôleur Midi (Maschine MKII de chez native instrument.)

La finalité du projet serait de pouvoir utiliser cet outil pour composer une performance live et permettre par la suite de devenir un outil de performance selon les différentes compositions possibles.

Sachant que la synthèse est modulaire, il suffit de changer le séquenceur et les réglages.

Je vais également essayer de pousser dans cette direction le développement, afin que l'outil soit facile d'accès avec un design simple pour que d'autres musiciens puissent projeter de la vidéo interactive pendant leur performance live.

Analyse des besoins :

Je vais utiliser **Pure Data** pour réaliser ce projet, car cette plateforme, déjà modulaire dans sa pensée de programmation graphique, me permettrait de réaliser tous les différents modules audio sans beaucoup de difficulté.

Utilisation de la librairie **Automatonism 2.0 de Johan Eriksson**

Concernant la partie vidéo, la manière la plus simple de traitement de ce module-là serait avec à la librairie **Ofelia - OpenFrameworks**

Pour d'autres besoins, comme l'addition de certains outils d'acquisition de données Midi, je suis en train d'évaluer encore mes besoins, mais sachant que cette partie est la fin de la conception d'architecture, je pense me pencher sur cette question plus tard.

Concernant le contrôle sur la plateforme Arduino, j'ai déjà les deux librairies qui conviennent à cette tâche : **Comport** et **Mr.Peach**.

Pour la conception de l'architecture de la patch sur **pure Data**, je pense Utilisé plusieurs objets PD (**Automatonism 2.0 de Johan Eriksson**), qui comporteront eux-mêmes des objets PD (secondaires, servant la programmation de chaque module).

Pour se donner une idée des différents modules principaux suivis de leurs modules internes secondaires :

Synthétiseur modulaire :

- OSC 1 : (frequency, fine, tune, square, sine, triangle, saw, sub) frequency modulation?
- OSC 2: (frequency, fine, tune, square, sine, triangle, saw, sub) frequency modulation?
- OSC 3 : (frequency, fine, tune, square, sine, triangle, saw, sub) normal
- Modulation: (triangle, square, sine) même chose qu'un LFO
- Filtre : (cut off frequency, resonance, HP, BP, LP, receives signal for mod)
- Mixer: (vol channel 1, vol channel 2, vol channel 3)
- ADSR : (attack, decay, sustain, release, receive signal for mod)
- AMP : (vol général)
- Signal out: (1, 2, 3)

Séquencer 16 pas pour la synthèse de 3 pistes :

- Tempo : BPM
- Transport : play, stop
- Grille de temps (step size) : 1/4, 1/8, 1/16, 1/32
- Start step : 1-16 (possibilité d'afficher les notes sur chaque pas, mais également un contrôle de position classique.)

Module de traitement de la vidéo :

- Une fenêtre qui affiche la vidéo grâce à Ofelia, chaque modules que j'ai crée et modifié de la suite d'exemple de la librairie va générer des formes différentes et le son interviendra sur chaque élément visuel

Interface utilisateur (pure data) :

Je suis en train de concevoir l'architecture à l'aide d'un schéma pour me donner une idée de ce que peut rendre l'interface numérique au niveau des contrôles.

Je préfère commencer à structurer l'interface maintenant pour m'éviter des chemins de programmation trop longs entre mes objets, même si ce n'est pas une priorité pour le moment.

Acquisition de connaissances :

Synthétiseur modulaire :

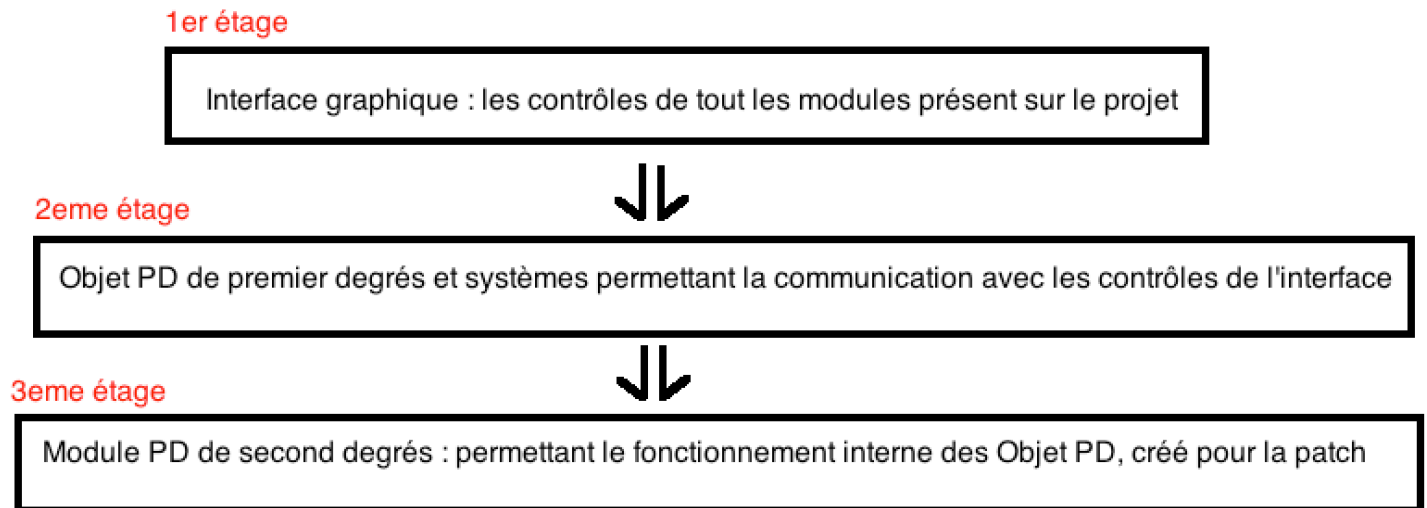
- Liste d'objet de PD :
http://blazicek.net/list_of_pure_data_objects.html
- Exemple de synthétiseur modulaire :
http://overdrive.music.free.fr/03patchwork_synth.php
- Exemple de synthétiseur modulaire utilisant pure data comme plateforme:
<http://www.monologx.com/xodular/>
- Exemple de synthétiseur modulaire utilisant pure data comme plateforme:
<https://www.automatonism.com/the-software/>

Module de traitement de la vidéo :

- *Librairie traitement video*
http://tot.sat.qc.ca/logiciels_pixeltango.html
- *Librairie générative video*
<https://puredata.info/downloads/gem/>
- *Librairie générative video*
<https://forum.pdpatchrepo.info/topic/11259/ofelia-pd-external-library-written-with-openframeworks>

Interface utilisateur (côté Pure Data) :

Modèle :



Méthodes :

1 : Synthétiseur modulaire :

« Utilisation des modules de la suite Automatonism 2.0 de Johan Eriksson »

Ce référé a la documentation pour les nouveaux utilisateurs:

<https://www.automatonism.com/synth-recipes/>

Vous pourrez vous construire votre propre version modulaire.

Les modules comprendrons chacun des controls distinctif.

- OSC 1 : (frequency, fine, tune, square, sine, triangle, saw, sub)
- OSC 2: (frequency, fine, tune, square, sine, triangle, saw, sub)
- OSC 3 : (frequency, fine, tune, square, sine, triangle, saw, sub)

pour chacun des paramètre il y sera associé un *slider* sur l'interface graphique de pure data, ce *slider* permettra de visualiser l'interaction avec le contrôleur MIDI.

pour mettre en fonction les 3 oscillateurs, il y aura un objet OSC de ma conception, qui sera relié a l'interface de contrôle avec un *throw signal*.

Pour ce qui est du *DAC*, je vais le placé pour l'instant au plus au niveau de la surface graphique car je veux avoir un contrôle dessus vue la quantité d'objet que contiendras mon projet.

- *Modulation: (triangle, square, sine) même chose qu'un LFO*

Pour cet objet je vais créé trois commutateur qui permettons de sélectionner le type d'onde que l'on désirs. la modulation sera composé d'un objet qui se trouvera dans l'objet OSC 1, 2 et 3. le contrôle de cette modulation se trouveras dans le *1er étage* (voir schéma 1) .

« **Quelque modification depuis mais reste le même principe de contrôle** »

2 : Patch Visuel :

Ofelia avec l'utilisation et la modification des exemples pour une Compréhension plus simple pour les Users

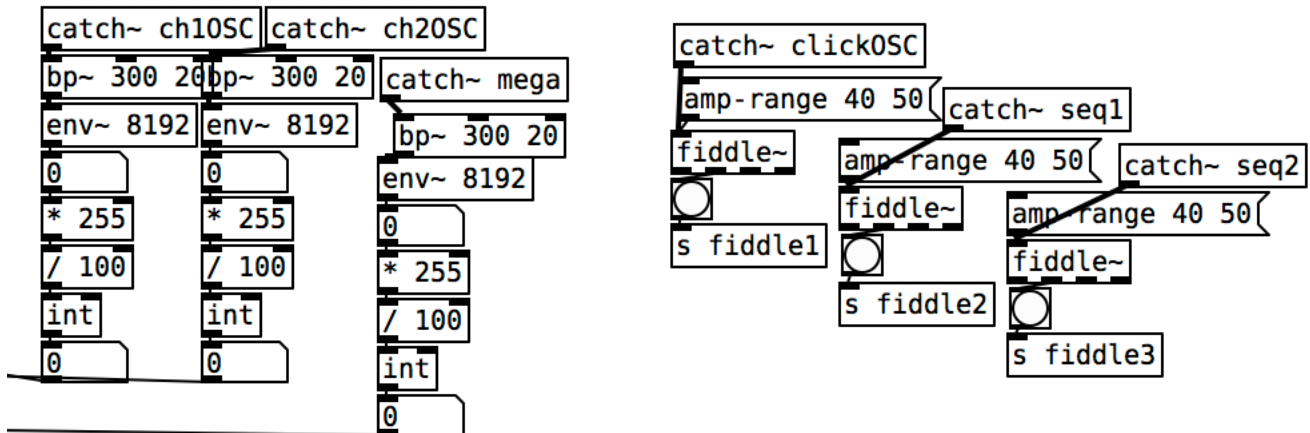
The library includes the following features:

- interactive output window
- various getters and event listeners
- 2d/3d shapes drawing
- image and font loading
- camera, lighting, material
- framebuffer object
- various data types (vec3f, color..)
- various utilities to speed up development (new expr, counter..)
- bandlimited oscillators and resonant filters

« *Listes des Objets présent dans la librairie* »

3 : Interactivité:

Utilisation des objets fiddle~ et env~ qui vont permettre de faire de la détection d'attaques ainsi que du filtrage de certaine fréquence et d'un Threshold.



« Capture d'écran de la patch audio, montrant les objets interactifs »

Mise à jour a venir:

- Ajout des commandes MIDI pour pouvoir contrôler la patch audio et visuel avec un contrôleur **Maschine MKII** de chez **Native Instrument**.
- Ajout de La boîte à rythme **Roland TR8** pour agrémenter la synthèse avec des éléments rythmiques.

Remerciement :

Je voudrais remercier mon Professeur, Olivier Belanger pour ses conseils tout au long du projet.

Johan Eriksson, créateur de tout les modules de synthèse ainsi que de la librairie Automatonism 2.0

Les créateurs de la librairie Ofelia.